

TDK-dolgozat

Dénes Botond

Vízközeli hulladéklerakók megbízható detektálása multispektrális műholdfelvételek segítségével

EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

PROGRAMOZÁSELMÉLET ÉS SZOFTVERTECHNOLÓGIAI TANSZÉK



Szerző:

Dénes Botond
programtervező informatikus MSc
2. évfolyam

Témavezető:

Cserép Máté
egyetemi tanársegéd

Budapest, 2024

Tartalomjegyzék

1. Bevezetés	3
1.1. A kutatólabor eddigi eredményei	4
2. Kapcsolódó kutatások	5
2.1. Hulladékdetektálási módszerek	5
2.2. Döntési fa	11
2.3. Random Forest	12
2.4. PlanetScope	13
3. Elemzés és tervezés	15
3.1. Kutatási cél	15
3.2. Műhold specifikációk	15
3.3. Használt indexek	15
4. Betanítás	18
4.1. Tanítóadatok	18
4.2. Tanítási paraméterek	19
4.3. Főkomponens analízis (PCA)	20
4.4. Nyári és téli adatokra való lebontás	22
5. Tesztelés	24
5.1. A teszthalmaz	24
5.2. Teljesítmény mérési módszerek	26
6. Megvalósítás és alkalmazás	28
6.1. A meglevő alkalmazások bővítése	28
6.1.1. Az asztali alkalmazás bővítése	28
6.1.2. A szerveralkalmazás bővítése	28
6.2. A Tiszta-Tisza alkalmazás	29

6.2.1. A képfeldolgozás gyorsítása	30
6.3. Közös metszet	33
7. Összefoglalás és eredmények	36
Köszönetnyilvánítás	37
A. Szimulációs eredmények	38
Irodalomjegyzék	39
Ábrajegyzék	44
Táblázatjegyzék	45
Algoritmusjegyzék	46
Forráskódjegyzék	47

1. fejezet

Bevezetés

szinkronizálni az
absztrakttal

A hulladékszennyezés komoly problémát jelent a természet számára [1]. Emiatt számos szervezet mozdul abba az irányba, hogy tisztábbá tegye a bolygónkat. Egy ilyen szervezet a PET Kupa, akik folyómenti hulladékgyűjtéssel foglalkoznak elsősorban Magyarországon, de figyelmük kiterjed a szomszédos országokra is. Az egyik nagy kihívás a szemétgyűjtésben a hulladékkal szennyezett területeknek a hatékony megtalálása. Sok erőforrást igényel a hulladéklerakók megtalálása a folyók mentén, hiszen sokszor járművel kell valakinek végig haladnia egy hosszabb területen, azért, hogy felmérje, hogy hol van hulladék. A folyók árterén elhelyezett hulladékok még nagyobb problémát jelentenek, hiszen dagály idejében a hulladékot elmosa a víz és ez a folyó további szakaszaira lesz szétszórva miközben nagy károkat okoz a folyó élővilágának, illetve szennyezi a folyóvizet [2, 3]. Emiatt szükségünk van olyan eszközökre, melyekkel hamar lehet detektálni a szennyezett területeket, hogy ezeket minél hatékonyabban meglehessen tisztítani. Ennek fényében a PET Kupa felkereste az egyetemünket azzal a kéréssel, hogy olyan eszközöket fejlesszünk le, melyek automatikusan képesek lesznek hulladékot detektálni a folyók mentén.

A dolgozatomban bemutatok egy Random Forest modellt [4], mely a kutatólaborban már lefejlesztett modellre épül [5]. A bemutatott modell javít a korábbi megoldás pontosságán, illetve nagyobb megbízhatósággal találja meg a hulladékot a folyókon és a folyók mentén. A modell eredményei integrálásra kerülnek a Tiszta Tisza webalkalmazásba, ahol több napon keresztül történő detektálás eredménye lesz összesítve és megjelenítve a felhasználók számára. Ezen felül a dolgozatban tárgyalni fogok más kutatást is, mely a hulladékdetektálás problémájával foglalkozik. Ezen kívül kiegészíttem a kutatólabor meglevő szoftveres eszközeit annak érdekében, hogy a laborban zajló

Hivatkozni a
weboldalra

munka gördülékenyebb legyen.

A kutatás hozzáadott terméke egy olyan adathalmaz, mely alkalmas más hulladékdetektálási modellek betanítására is. Az adathalmaz elsősorban szárazföldi romániai hulladéklerakókról készített PlanetScope műholdfelvételeket tartalmaz, melyek kézzel voltak annotálva. Az adatok georeferálva vannak, így ezeket könnyen meg lehet vizsgálni, illetve ki lehet egészíteni.

Továbbá bemutatom, hogy milyen módszerekkel próbáltam tovább javítani a modell eredményein. Ilyen módszer például a főkomponens analízis, a képnormálizálás, illetve az évszakokra bontás.

1.1. A kutatólabor eddigi eredményei

A térinformatikai kutatólaborban már fejlesztésre került egy szerveralkalmazás, mely minden nap a Planet szervereiről letölти a legfrissebb felvételeket a vizsgált területekről, és lefuttatja ezeken a képeken az akkori modellt. Ezen felül készült egy webalkalmazás is, ami erről a szerverről letölти az eredményeket, és megjeleníti ezeket, összehasonlításra. A kutatólabor rendelkezik egy asztali alkalmazással is, mellyel hatékonyan elő lehet állítani tanítóadatokat. A kutatásom elősegítéséhez ezeket az alkalmazásokat használtam, illetve bővítettem a 6.1 fejezetben leírtak szerint.

2. fejezet

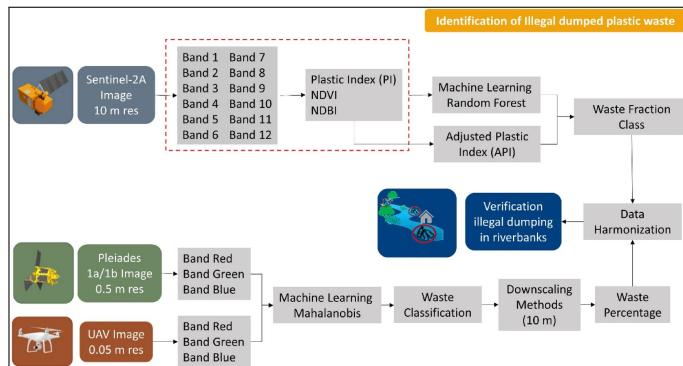
Kapcsolódó kutatások

2.1. Hulladékdetektálási módszerek

A hulladékdetektálás témája viszonylag új, és számos fejlemény történt az évek során, tekintve, hogy a légi és műholdfelvételek minősége is nőtt. Emiatt kevés olyan kutatás létezik, mely pontosan azzal a problémával foglalkozik, amivel ebben a dolgozatban foglalkoztam, de az érintőleges kutatások is olyan módszereket, ötleteket mutatnak be, melyeket érdemes megfontolni ebben a kutatásban is. Az én kutatásom a Térinformatikai kutatólabor munkájára épül [5], ahol egy Random Forest modell került betanításra, hulladékdetektálás céljából. A kutatásban PlanetScope és Sentinel-2 műholdfelvételeket használtak. Ez a cikk rakta le az alapjait az én kutatásomnak is, melyben ezeken az eredményeken javítok. A cikkben további lehetséges munkaként említésre kerül a modell több adattal való tanítása, illetve a képfeldolgozás gyorsítása. A dolgozatom minden feladattal foglalkozik, A dolgozatomban csak a PlanetScope felvételek kerülnek felhasználásra, mivel a magasabb felbontású felvételek könnyebben lehetővé teszik a tanítóadatok előállítását, hiszen jobban lehet látni a hulladéklerakókat rajtuk.

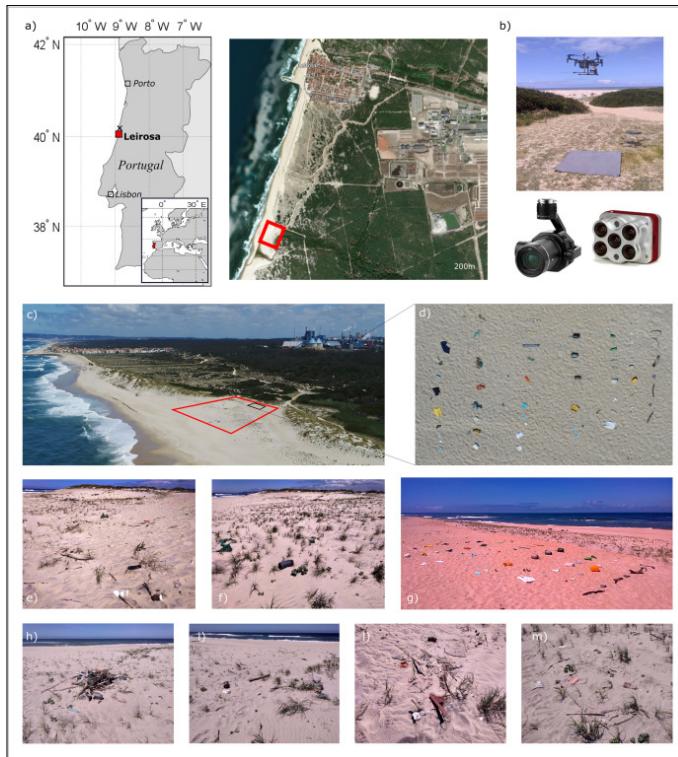
[6]-ben Sentinel-2 műholdfelvételeken tanítottak be egy Random Forest modellt azzal a céllal, hogy egy indonéziai folyóban detektáljanak hulladékot. A cikkben bevezetik az "Adjusted Plastic Index"-et, mellyel a vegetáció, föld és épületek közötti zajt csökkentik. Ennek az indexnek a kiszámításához a Sentinel-2 műhold piros, közeli infravörös (NIR), illetve rövid hullámhosszú közeli infravörös (SWIR) sávokat használták fel. Validációjának Pleiades műholdképeket és drónfelvételeket klasszifikáltak Mahalanobis távolság gépi tanulási módszerrel (2.1 ábra). A módszer növényzetben és vízen rendre 88%, illetve 85% pontosságot ért el és épületeken, törmeléken és földön rendre 62%, 53%,

illetve 21% pontossággal tudta a hulladékot detektálni. A cikk szerint az utóbbi három adattípuson azért visszafogottabbak az arányok, mert a spektrális értékei az épületeknek, a földnek és a törmeléknak nagyon hasonlítanak.



2.1. ábra. Hulladékdetektálás "Adjusted Plastic Index", Random Forest és Mahalanobis távolság segítségével [6]

[7]-ben Spectral Angle Mapping módszert alkalmaztak multispektrális drónfelvételeken, egy Portugál tengerparton. A célja a kutatásnak az volt, hogy a tengerparton kimosott hulladékot detektálják és klasszifikálják. A módszer alkalmazásához referencia értékeket állítottak elő azzal, hogy elhelyeztek különböző anyagokból álló hulladékot a homokba, és ezekről drónfelvételt készítettek (2.2 ábra). Ezzel a módszerrel képesek voltak detektálni és klasszifikálni nem csak homok fölött található hulladékot, hanem a homokban félig elásott hulladékot is. A 472 kézzel előállított tesztadatból volt a 268 True Positive (57% összesen), 96 volt a False Positive és 204 volt a False Negative.

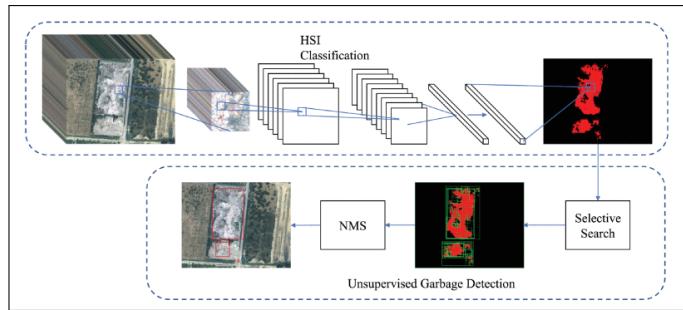


2.2. ábra. Spectral angle mapping referencia adatainak előállítása [7]

[8]-ben mezőgazdasági hulladékdetektálásra használtak egy Support Vector Machine modellt, Landsat 8 műholdfelvételeken. A szenzor Kék, Zöld, Piros, NIR, SWIR 1, SWIR 2 és CIRRUS sávját használták a tanítóadatok és tesztadatok előállítására. Ezután véletlenszerűen szétválasztották az adatokat tanítóadatokra és tesztadatokra. A következő osztályokra bontották az adatokat: Hálók, műanyag takarók, föld, növényzet, gyümölcsök kertje, olajfás kert, város, fa, fás föld. A modell a tesztadatokat összességében 94%-os pontossággal tudta klasszifikálni, ahol a legrosszabb arányokat az olajfás kert érte el 77.78%-os pontossággal.

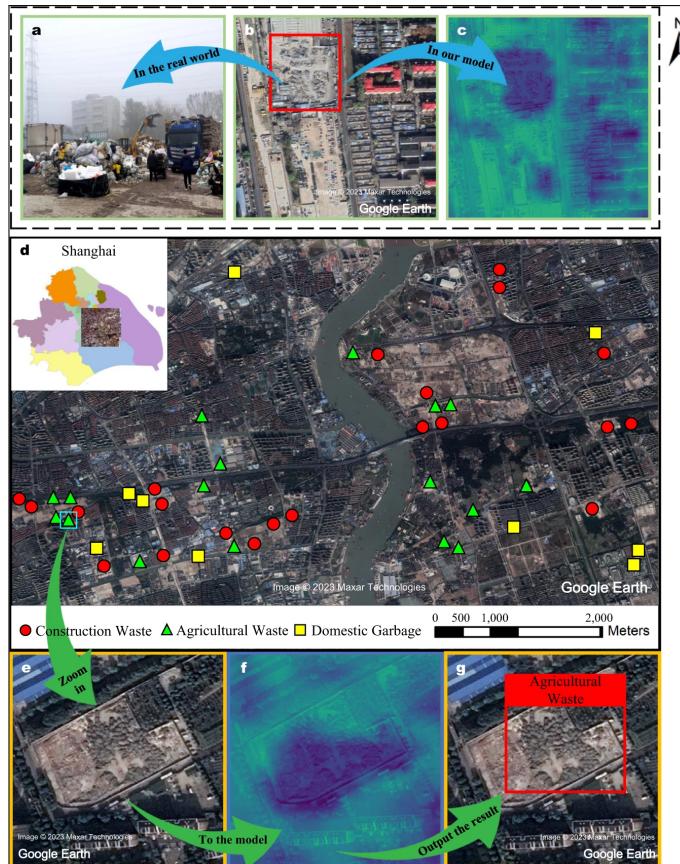
[9]-ben a hiperspektrális adatokon tanítottak be egy felügyelt és egy felügyeletlen gépi tanulási módszert. A működési elv az, hogy a felügyelt módszerrel klasszifikálják a hulladékkel szennyezett területeket, míg a felügyeletlen módszerrel megbecsülik a hulladékkel szennyezett terület kerületét (2.3 ábra). Ők 99.89%-os kappa együtthatóval tudtak hulladékot detektálni .

Átnézni
mégegyszer
a cikket és
átvizsgálni
az adatok
helyességét



2.3. ábra. Az MSCNN működési elve [9]

[10]-ben megvizsgálják azt, hogy mennyivel lesz hatékonyabb a hulladékkezelés a manuális vizsgálat helyett, ha egy mély tanulás alapú hulladékdetectálási módszert használnak a hulladékkezeléshez magas (0.3m-1m per pixel) felbontású műholdfelvételeken. Négy osztályba bontják a hulladékot: háztartási hulladék, mezőgazdasági hulladék, építkezési hulladék, lefedett hulladék. A modell képes 98%-át detektálni a hulladéklerakóknak a teszthalmazban, illetve az automata detectálás segítségével több, mint 96.8%-al csökkentették azt az időt, ami a hulladékdetectálás vizsgálatára kellett. Ráadásul a modell könnyen telepíthető egy laptopra, és 30 másodperc alatt le tudja futtatni a modellt a 162 négyzetkilóméteres tesztadathalmazon. A betanításhoz minden össze 2500 hulladéklerakót annotáltak világszerte kézzel 4800 négyzetkilóméteren keresztül. A 2.4 ábrából látható a módszer működési elve: a modell a spektrális értékek mellett a hulladéklerakó alakját is figyelembe veszi. A megfelelő magabiztoságú területek hulladéklerakóként lesznek címkézve. A modellt betanító és validáló kód, illetve az adathalmaz, mellyel az eredmények reprodukálhatóak elérhetőek a cikkből.



2.4. ábra. Az BCA-Net működési elve [10]

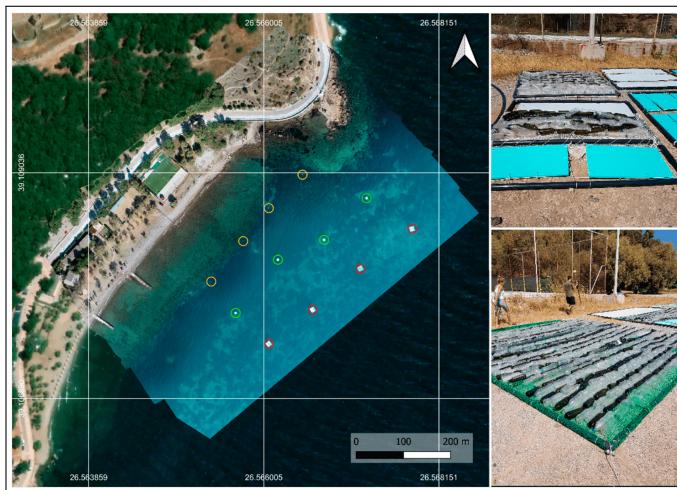
[11] egy konvolúciós neuronhálót használ, mely 94.5%-os átlagos pontossággal és 88.2%-os F-Score-al rendelkezik. A tanító adathalmaz 3000 ortofotóból áll, melyek 20cm per pixel felbontással rendelkeznek és a vörös, kék és zöld tartományokat tartalmazzák. A felvételeket szakértők annotálták, kézzel. A modell a ResNet50 [12] hálóra épül és a Feature Pyramid Network [13] architektúrával van kiegészítve. A modell figyelembe veszi a hulladéklerakó alakját és kontextusát, mely segít a hulladékdetectálás pontosságának a növelésében.

[14]-ban két Random Forest algoritmust tanítottak be kerekek és műanyagok detektálására, Skóciában. A kutatásukban A Copernicus Sentinel-1 és Sentinel-2 multispektrális műholdfelvételeit használták. A tesztelés során a modellek 211 kerék és műanyag alapú hulladéklerakót találtak, rendre 87.5% és 84% pontossággal. A kutatásukban az NDVI, SAVI, illetve NDWI2 indexek kerülnek tárgyalásra és kiszámításra.

[15] tengeri hulladékdetectálással foglalkozott Görögország területén levő tengerpartokon: egy felügyelt (Light Gradient Boosting Model) és egy felügyeletlen (K-Means) gépi tanulási algoritmust alkalmaztak PRISMA hiperspektrális

műholdfelvételeken, mellyel átlagosan 96%-os pontosságot értek el. A tanítóadatokat kontrollált környezetben állították elő (2.5): különböző környezeteket szimulálva vízre helyeztek különböző anyagokból álló hulladékokat, melyekről műholdfelvételek készültek. Ennek köszönhetően pontosan elő tudták állítani a tanítóadatokat.

átfogalmazni,
mégegyszer
elolvasható
cikket



2.5. ábra. A tengeri hulladékdetektáláshoz való tanítóadatok előállítása [15]

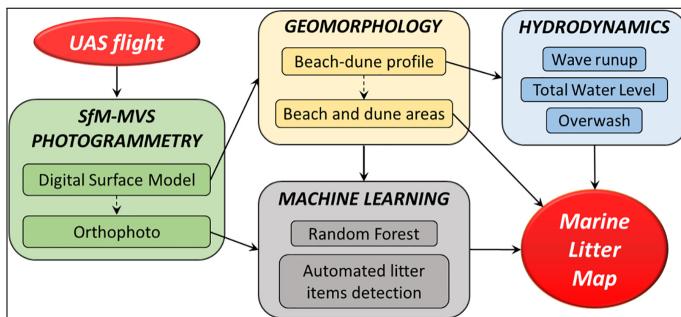
[16]-ban tengerparti hulladékdetektálással foglalkoztak, úgy úszó hulladékkal, mint partra kimosott hulladékkal. A kutatásban bemutatnak egy APLASTIC-Q konvolúciós neuronhálót, mely 5 pixel per cm felbontással rendelkező felvételeket klasszifikált Kambodzsa területén. A modell két fő komponensből állt: egy műanyag-hulladék detektálóból (PLD-CNN) és egy műanyag-hulladék osztályozóból (PLQ-CNN). A PLD-CNN megkülönböztette a vizet, homokot, növényzetet és műanyag alapú hulladékot 83% átlagos pontossággal, illetve megállapította, hogy az adott terület mennyire hulladékos. PLQ-CNN megkülönböztette a különböző hulladék típusokat átlagosan 71%-os pontossággal: ilyenek például az üvegpalackok, cipők, textil anyagok stb. A PLD-CNN betanításához 5515 darab $100 \times 100 \times 3$ pixeles csempét használtak, míg a PLQ-CNN betanításához 4828 darab $50 \times 50 \times 3$ pixeles csempékre vágták a felvételeket.

[17]-ban gépi tanulást, geomorfológiát, fotogrammetriát és hidrodinamikai modellezést kombinálnak össze azzal a céllal, hogy homokos partok környékén, illetve homokdombokon detektáljanak hulladékot drónfelvételek segítségével (2.6 ábra).

A geomorfológiai vizsgálat segítségével megtudták különböztetni a partokat és a homokdombokat, mellyel optimizálni tudták a hulladékdetektálást. A gépi tanulási modell egy Random Forest volt, mellyel a homokban található hulladékot detektálták. A hidrodinamikai számolások segítségével meg lehetett becsülni, hogy hol kerülnek a

jobb ábrát
keresni

hulladékok kimosásra, illetve a parton található hulladékok mennyit fognak ott maradni. Összességében ezzel a módszerrel a hulladékdetektálás 75%-os F-Score-al sikerült.

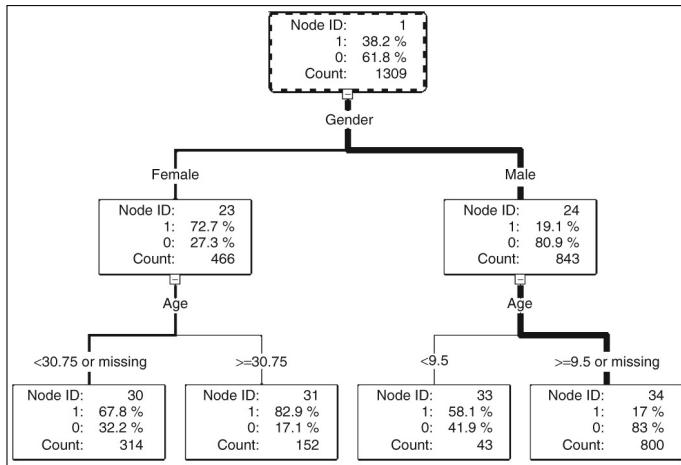


2.6. ábra. A gépi tanulás, geomorfológia, fotogammetria és hidrodinamikai modellezés használása hulladékdetektálásra [17]

[18]-ben konvolúciós neuronhálót (CNN) használtak, hogy a Szenegál folyó mentén hulladékot detektáljanak Saint-Louis-ban. A kutatáshoz magas felbontású drónfelvételeket használtak. Drónfelvételeket készítettek 5, 10, 30, illetve 300 méter magasságon, úgy, hogy a látási szög merőleges volt a földre. A 10 méter és 30 méteres felvételeket használták a modell betanítására. A kiválasztott kutatási területnek az előnye, hogy változatos környezetben található, így modell betanítására és tesztelésére is alkalmas. A modell átlagos pontosságát nem mérték, mivel nagy eltérések voltak a vizsgált régiók függvényében. A cikkben említi, hogy a modell pontossága lényegesen növelhető, ha több adattal kerül betanításra.

2.2. Döntési fa

A döntési fák általános célú predikcióra és osztályozásra használt eszközök [19]. Az előnyeik közé tartozik az, hogy tekintve az egyszerű struktúrájukat, könnyen értelmezhetőek és rugalmasak. A 2.7 ábrából látható a döntési fa működési elve.



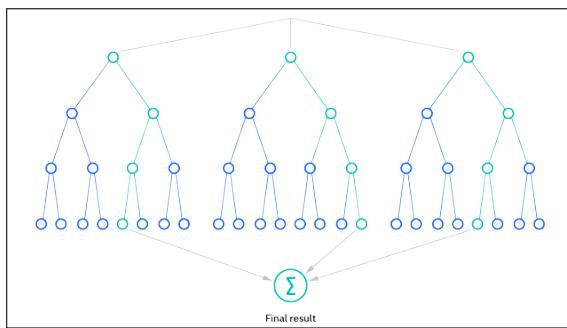
2.7. ábra. A döntési fa működési elve [19]

A 2.7 ábrán látható döntési fa a Titanic utasairól készített túlélési kutatás adatai alapján épült [20]. A fa csúcsain található 1-es címke képviseli a túlélők arányát, míg a 0-ás címke az elhunytak arányát. Mivel a 0 és 1 osztályok közül választhat a döntési fa, ezért ez egy osztályozó döntési fa. Ezen felül a csúcsban el van tárolva az is, hogy hány mérést tartalmaz. A gyökérkötél tartalmazza az összes mérést. Ahhoz, hogy leágazzunk egy csúcsból, meg kell határoznunk a közvetlen gyerekcsúcsok "bemeneteit": ezek olyan mezők az adathalmazban, melyek a legjobban leírják a változékonyságát az adatoknak az adott szinten. A legalkalmasabb bemenetek meghatározása egy nyitott kutatási téma. Ebben a példában a gyökérkötél közvetlen gyerekeinek a bemenete az utas neme. A következő szint bemenete pedig az utas kora. Így például a következő információkat olvashatjuk le a döntési fáról: a 30.75 év fölötti nőknek volt a legnagyobb túlélési esélyük, míg a 9.5 fölötti vagy ismeretlen életkorral rendelkező férfiaknak volt a legkisebb túlélési esélyük. Így tehát ha meg szeretnénk állapítani, hogy mennyi lett volna a túlélési esélye egy 35 éves férfinek, akkor látnánk azt a döntési fa alapján, hogy ez 17% lenne.

2.3. Random Forest

A 2.2 fejezetben tárgyalt döntési fa egyik hátránya az, hogy hajlamos a túltanulásra. A Random Forest egy felügyelt gépi tanulási módszer, mely magas pontossággal tud rátanulni a tanítóadatokra (túltanulás nélkül), és jól kezeli a zajt [4]. A módszer több, véletlenszerűen paraméterezett döntési fa felépítéséből kapta a nevét: miután felépítettük ezt a "döntési fa erdőt", az adatokat úgy lehet osztályozni, hogy egy többségi szavazást hajtunk végre az összes fa eredménye szerint (2.8 ábra). Ennek a megközelítésnek

köszönhetően az egyes túltanult döntési fák kiegyenlítik egymást, így a teljes erdő nem lesz túltanítva. A fák felépítéséhez több stratégia létezik, és ezen stratégiák alapján lehet finomhangolni a modell pontosságát és méretét. Az utóbbit fontos szem előtt tartani, tekintve arra, hogy elég sok tanítóadat esetén a modell mérete lényegesen megnőhet helyes paraméterezés hiányában. Az ilyen paraméterek például a fák maximum mélysége, a fáknak átadott részadathalmaz dimenziói, egy csúcs kettéválasztásának a kritériumjai, a tanítóadatok súlyai stb. A kutatásomban ezt a modellt tanítom be, illetve paraméterezem azzal a céllal, hogy megbízható klasszifikációt tudjon biztosítani. A modell minden képkockát osztályozni fog, így a bemeneti adatok az adott terület spektrális sávjai, illetve indexei.



2.8. ábra. A Random Forest működési elve [21]

2.4. PlanetScope

A Planet Labs rendszeresen műholdakat küld az űrbe azzal a céllal, hogy minden nap készüljön műholdfelvétel a Földről [22]. A projekt jelenlegi állapotában már ott tartanak, hogy majdnem napi szintű felvételeket készítenek a Föld felszínéről több, mint 430 Dove és SuperDove műhold segítségével [23]. A legújabb műholdcsaládjuk a SuperDove műholdakból áll. Ezek a műholdak a PSB.SD nevű műszert használják felvételek készítésére. A műszer a 47 megapixeles "PSBlue" szenzort használja. Ezekkel a műholdakkal 2020. március közepe óta készítenek felvételeket, melyek a vörös (red), zöld (green), kék (blue), közeli infravörös (near infrared vagy NIR), illetve az úgynévezett "red edge", "green I", "coastal blue", és sárga (yellow) sávokat is tartalmazzák. Egy csempé körülbelül $32.5 \times 19.6 \text{ km}^2$ területet fed le [24], körülbelül 3 méter per pixel felbontással. A 2.1 táblázatból látható, hogy a PSB.SD műszer egyes sávjai milyen hullámhosszal rendelkeznek. A hullámhossz mellett jelenített "fwhm" a "Full width at half maximum" értéknek a rövidítése.

Sáv azonosító	Sáv neve	Hullámhossz (fwhm)
1	Coastal Blue	443 (20)
2	Blue	490 (50)
3	Green I	531 (36)
4	Green	565 (36)
5	Yellow	610 (20)
6	Red	665 (31)
7	Red Edge	705 (15)
8	NIR	865 (40)

2.1. táblázat. A PSB.SD műszer hullámhosszai [24]

3. fejezet

Elemzés és tervezés

3.1. Kutatási cél

A cél az, hogy a kutatás során szerzett modell megbízhatóan detektáljon hulladéklerakókat általánosan folyók mentén. Ehhez a false positive arányok minél kisebbek kell legyenek, míg a true positive arányok minél nagyobbak. Ugyanakkor nem jelent ugyanakkora problémát egy false negative, mint egy false positive, mivel a false positive eredmények fölöslegesen rossz irányba küldhetik a folyómentő csapatot. A kutatólabor 2023-as cikkjében bemutatott modell (továbbiakban meglevő modell) egyik problémája a nagy false positive arányok voltak. A modell a pusztazámi hulladéklerakóról, illetve a kiskörei víztárolóról szerzett adatokkal volt betanítva. Ezért érdemes első körben egy nagyobb adathalmazzal betanítani a modellt.

3.2. Műhold specifikációk

Az új Random Forest modell a PlanetScope műholdakra van specializálva, azon belül is a legújabb PSB.SD szenzorokra. A modell a Vörös (Red), Kék (Blue), Zöld (Green), és a közelí infravörös (NIR) sávokat használja. A PlanetScope műholdak körülbelül 3 méter/pixel felbontással rendelkeznek [24].

3.3. Használt indexek

A kutatás során felhasználom a kutatólaborban már számolt indexeket. Pontosabban a Plastic Index (3.1 képlet), Normalized Difference Water Index (3.2 képlet), Normalized

Difference Vegetation Index (3.3 képlet), Reversed Normalized Difference Vegetation Index (3.4 képlet), Simple Ratio (3.5 képlet) indexek kerülnek használatra [25, 5].

$$\text{Plastic Index (PI)} = \frac{\text{NIR}}{\text{NIR} + \text{Red}} \quad (3.1)$$

$$\text{Normalized Difference Water Index (NDWI)} = \frac{\text{Green} - \text{NIR}}{\text{Green} + \text{NIR}} \quad (3.2)$$

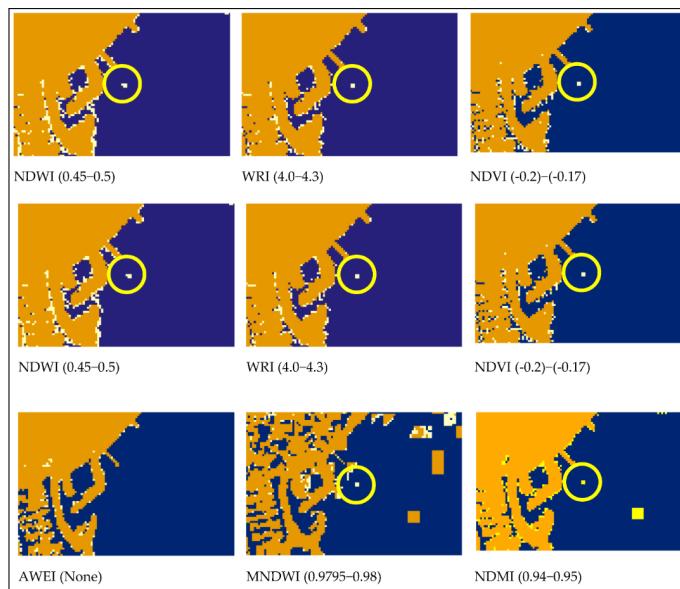
$$\text{Normalized Difference Vegetation Index (NDVI)} = \frac{\text{NIR} - \text{Red}}{\text{NIR} + \text{Red}} \quad (3.3)$$

$$\text{Reversed Normalized Difference Vegetation Index (RNDVI)} = \frac{\text{Red} - \text{NIR}}{\text{Red} + \text{NIR}} \quad (3.4)$$

$$\text{Simple Ratio (SR)} = \frac{\text{NIR}}{\text{Red}} \quad (3.5)$$

A Plastic Index, ahogy a neve is sugallja, egy olyan index, mely a vízen úszó hulladékok pixelein magas értékeket vesz fel, míg a vízpixeleken alacsony értékeket. A számításban kihasználásra kerül az, hogy a közeli-infravörös tartományban a hulladék sokkal jobban visszaverődik a vízhez képest, mint a vörös tartományban. Az NDWI a NIR és zöld sávokat használja fel a vízben található tárgyak, illetve növények kijelölésére [26]. Az NDVI -1 és +1 értékek között tartózkodik. Ha adott területen az NDVI negatív értéket vesz fel, akkor nagy az esélyünk arra, hogy a terület az víz. Ha 1-hez közelí értéket vesz fel, akkor nagy eséllyel növényzet borítja az adott területet. Az RNDVI ennek a -1-szerese. A Simple Ratio is a vörös és közeli infravörös tartományokat használja ki a vegetáció további detektálásához és osztályozásához.

[25]-ban összehasonlításra kerülnek ezek az indexek (3.1 ábra): Összekötöttek több műanyag palackot, és felhelyezték a vízre. Ezután Sentinel-2 felvételekkel összehasonlították a különböző indexek teljesítményét ennek a szigetnek a megtalálásában. A kutatás eredménye az volt, hogy a Plastic Index tudta a legjobban kiemelni a műanyag-szigetet.



3.1. ábra. Különböző indexek összehasonlítása egy mesterségesen előállított úszó műanyag-palack szigeten [25]

4. fejezet

Betanítás

4.1. Tanítóadatok

A betanításhoz 29 romániai hulladéklerakó és közvetlen környezete került a tanítóadatok közé, illetve a Kiskörei víztároló is. A romániai hulladéklerakókat egy helyi weboldalon lehet megtalálni, a hozzájuk tartozó koordinátákkal együtt [27]. Az ott bemutatott 46 hulladéklerakó közül 29 volt alkalmas tanításra: sok hulladéklerakó be lett tölve, vagy föld alatt működik. minden hulladéklerakóhoz letöltöttem egy-egy nyári+tavaszi, téli és őszi multispektrális műholdképet, melyeket kézzel annotáltam. A nyári és tavaszi képeket azért vontam egybe, mivel ezek hulladékdetectálás szempontjából hasonló adatokat eredményeztek. A tanítóadatok pixelenként vannak előállítva, így a végső adathalmaz 27 millió tanítóadatból (pixelből) áll. minden pixelhez hozzá van rendelve a vörös, kék, zöld, közel infravörös sáv, illetve a "PI", "NDWI", "NDVI", "RNDVI", "SR" indexek. Ezen felül minden pixel címkézve van a 4.1 táblázatban leírtak szerint. Nagyon fontos, hogy minél pontosabban meg lehessen állapítani, hogy a hulladéklerakókon mely területeken található hulladék, hiszen sok hulladéklerakón törmeléket is tárolnak, ezt általában egy külön területen. Emiatt a tanítóadatokat a következő módszerrel állítottam elő: Megvizsgáltam Google Maps segítségével [28], hogy az adott hulladéklerakónál hol tárolnak törmeléket és hol tárolnak műanyag alapú hulladékot. A Google Maps légi felvételei elég magas felbontással rendelkeznek ahhoz, hogy szemmel meglehessen különböztetni a műanyag alapú hulladékot a törmeléktől. A 4.1 ábrából látható, hogy még a törmelék inkább fehér színt tartalmaz, addig a műanyag alapú hulladék picit vörösebb, tekintve arra, hogy a műanyag sokszor színezve van. Emellett, a hulladéklerakókat nagyon konzervatívan jelöltettem ki: csak akkor jelöltettem be

egy területet, mint hulladékos terület, ha a felvételből egyértelmű volt, hogy az adott terület műanyag alapú hulladékkel szennyezett.

Címke azonosító	Címke neve	Címke magyarázat
100	Hulladék	Azon területek, melyeken hulladékot találtunk.
200	Víz	olyan területek, melyeken kizárolag vizet találtunk, általában folyók.
300	Legelők/Erdők	Zöld övezetből álló vad területek. Ezek lehetnek fák lombjai vagy füves zónák.
400	Mezők	Olyan földes területek, melyek meg vannak művelve, illetve ahol mezőgazdasági növények találhatóak, például gabonafélék.
500	Ismeretlen	Olyan területek, melyek a korábbi kategóriákba nem sorolhatók bele. Ilyenek az épületek, aszfaltozott utak, háztetők, mezei utak.

4.1. táblázat. A tanítóadatok címkei



(a) Műanyag alapú hulladék

(b) Törmelék

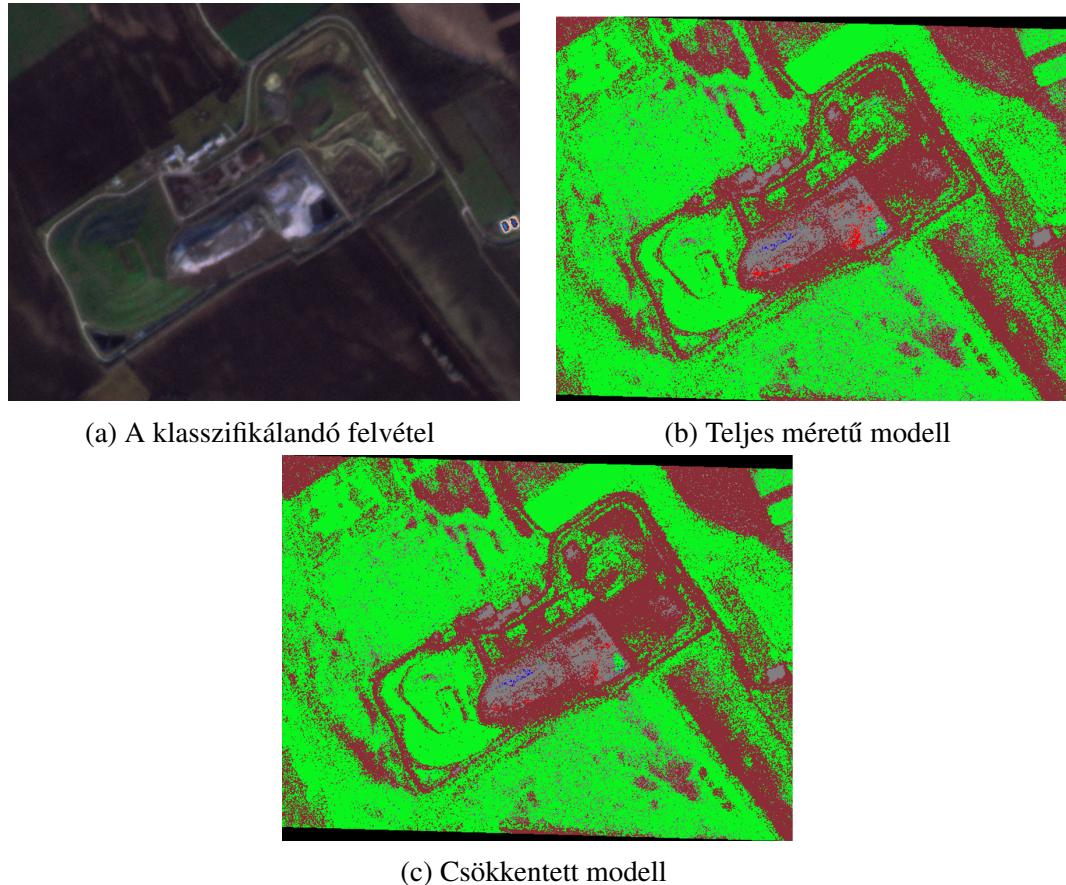
4.1. ábra. A műanyag alapú hulladék törmelék mellé helyezve. Forrás: Google Maps

4.2. Tanítási paraméterek

A nagy adathalmaz miatt a Random Forest modell is nagyon nagy lesz (körülbelül 14GB), ami egy nehezen kezelhető méret, így érdemes módosítani a modell paraméterein, hogy ez kisebb méretű legyen. A legjobb eredményeket azzal értem el, hogy a Random Forest fák méretét 20 mélységűre limitáltam. Ennek köszönhetően a modellek méretét

2GB-ra tudtam csökkenteni, és a 4.2 ábrából látható, hogy a csökkentett modellben

enyhén megnő a hulladékra vonatkozó false-negative ráta, míg a false positive arány nem nő, de cserében egy kezelhető méretű modellt kapunk.



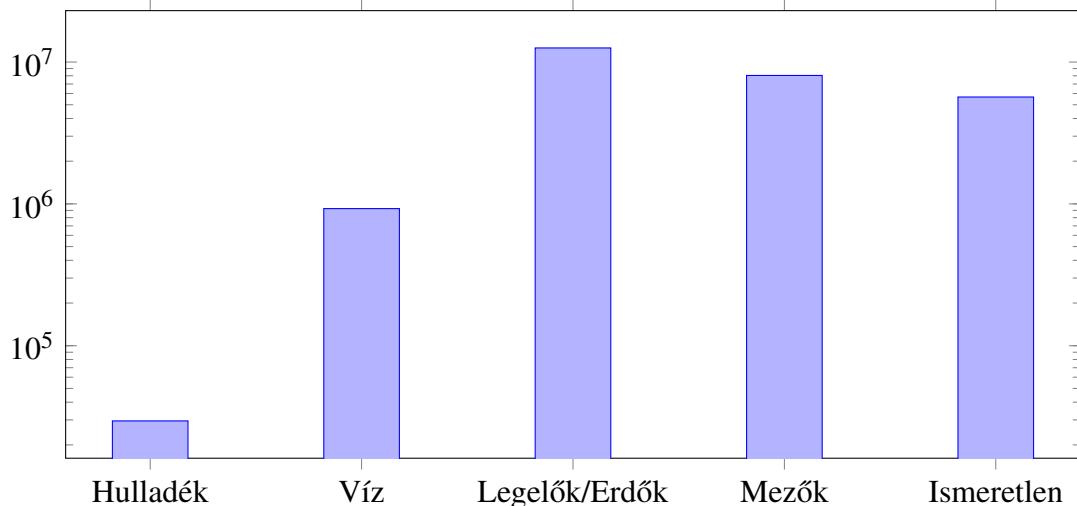
4.2. ábra. A csökkentett modell hasonlóan teljesít a teljes méretű modellhez

Továbbiakban felmerült az a probléma is, hogy a tanítóadatok nagyon aránytalanok voltak: A 4.3 ábrából látható, hogy nagyságrendekkel kevesebb adattal rendelkeztünk hulladékról, mint az összes többi adatról. Emiatt a modell nagyon sok false-negatívöt termelt. Ennek korrigálására súlyokat alkalmaztam a tanítóadatokra. A súlyok kiszámolásához az összes címkére a 4.1 képletet használtam.

$$\text{címke súlya} = \frac{\text{adathalmaz mérete}}{\text{címke darabszáma}} \quad (4.1)$$

4.3. Főkomponens analízis (PCA)

A modell méretének a csökkentésére megvizsgáltam a főkomponens analízis (PCA) alkalmazását is [29]. A módszer lényege az, hogy egy többdimenziós adathalmazból kivonja a legfontosabb információkat egy alacsonyabb dimenziószámú adathalmazba.



4.3. ábra. Az adatok közötti aránytalanság, logaritmikus skálázással

Az ötlet az volt, hogy a bemeneti adatok dimenziószámának a csökkentésével csökkeni fog a modell mérete, de érdekes módon a modell mérete nem csökkent a dimenziószám csökkentésével, helyette lényegesen megnőtt. Ezt az is tükrözi, hogy megnőtt átlagosan a Random Forest döntési fáinak a mérete, minél kevesebb dimenziószámú adatot kapott. A 4.2 táblázatból látható, hogy különböző főkomponenseknél mekkora volt átlagban a fák mérete. További vizsgálatok után kiderült, hogy hogyha kevesebb dimenziójú adatot adtam a modellnek, akkor a mérete lényegesen megnőtt. Ezen felül az is célja volt a PCA alkalmazásának, hogy a kiszámolt indexek információit megtartva, alacsonyabb dimenziószám segítségével a Random Forest modell hatékonyabban fogja majd feldolgozni a műholdfelvétteleket. [30] megmutatja, hogy egyes modelleken jobb klasszifikációt tudtak elérni több spektrális sávból álló adat esetén. Ennek az az oka, hogy ha igen sok a korreláció a különböző dimenziók között, akkor a modell rátanulhat a zajra. Mivel a modell által használt indexek között sok a korreláció, érdemes megvizsgálni a PCA-t olyan szempontból is, hogy esetleg javít-e a klasszifikációs eredményeken. A PCA alkalmazása a Random Forestre a következő lépésekkel áll:

Főkomponensek száma	Fák méretének mediánja
Főkomponensek nélkül (9 dimenzió)	71.5
5 főkomponens	71
4 főkomponens	73
3 főkomponens	85

4.2. táblázat. A döntési fák méretének a mediánja nem csökkent, amint a főkomponensek száma csökkent, cserében 3 főkomponensnél már nőtt.

1. A tanítóadatok standard skálázása (4.2 képlet).
2. A főkomponens analízis alkalmazása a tanítóadatokra.
3. A Random Forest modell betanítása a tanítóadatokon.

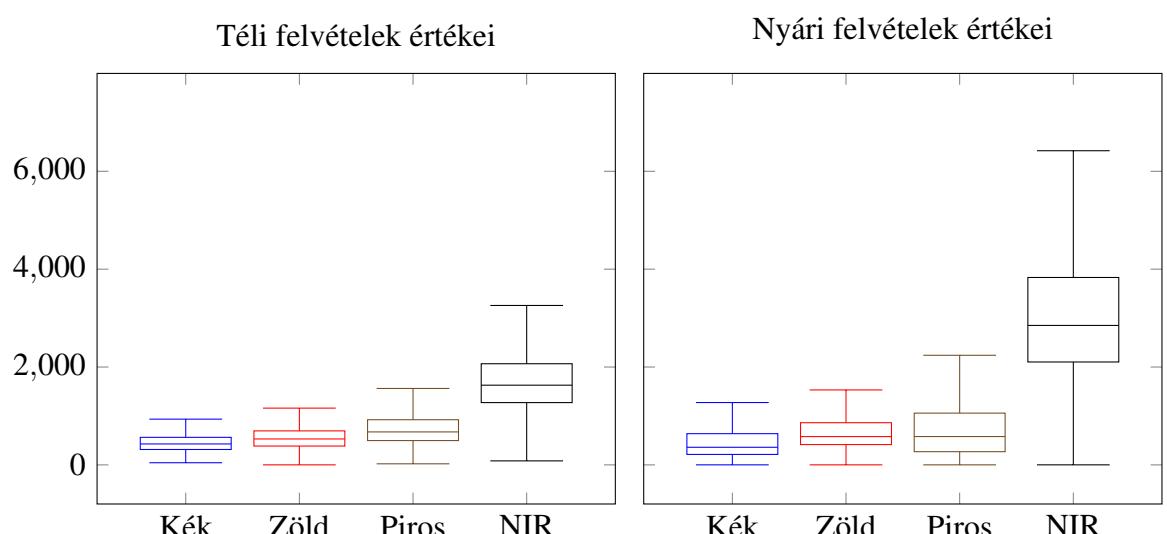
$$f(x) = \frac{x - \text{Average}}{\text{Standard Deviation}} \quad (4.2)$$

A modell tesztelésére is ugyanezeket lépéseket kell elvégezni. A főkomponens analízis használásához a scikit-learn [31] Python programcsomagot használtam.

4.4. Nyári és téli adatokra való lebontás

Alapértelmezetten a nyári és téli adatok között lényeges különbség tud lenni távérzékelés szempontból Közép-Európa területén: a téli időszakokban gyerebb a vegetáció, ködösebb a levegő, illetve a nap sem süt ugyanabból a szögből. Ez befolyásolhatja a modell pontosságát is az adott időszakban. A nyári időszakot márciustól októberig tartó időszakként definiáltam, és a téli időszak pedig novembertől februárig tart. Az időszakok aszerint vannak megválasztva, hogy mikor leveleznek ki, illetve hullatják ki a leveleiket a fák. Valóban, az októberi időszakban már inkább sárgásak lesznek a levelek, de az októberi tanítóhalmaz mérete önmagában igen kicsi érdemi tanításra. A 4.4 ábrából látható, hogy főleg a közel a infravörös (NIR) sávokon nagy eltérések vannak a nyári és téli felvételek között. Ennek fényében betanítottam külön egy nyári és egy téli modellt, melyek teljesítményét a ?? fejezetben részletezem.

fejezet
elkészítése



4.4. ábra. Nyári és téli adatok összehasonlítása

5. fejezet

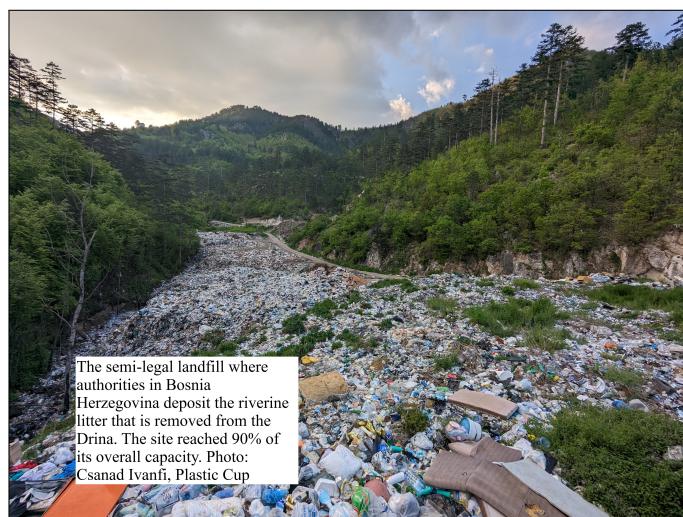
Tesztelés

5.1. A teszthalmaz

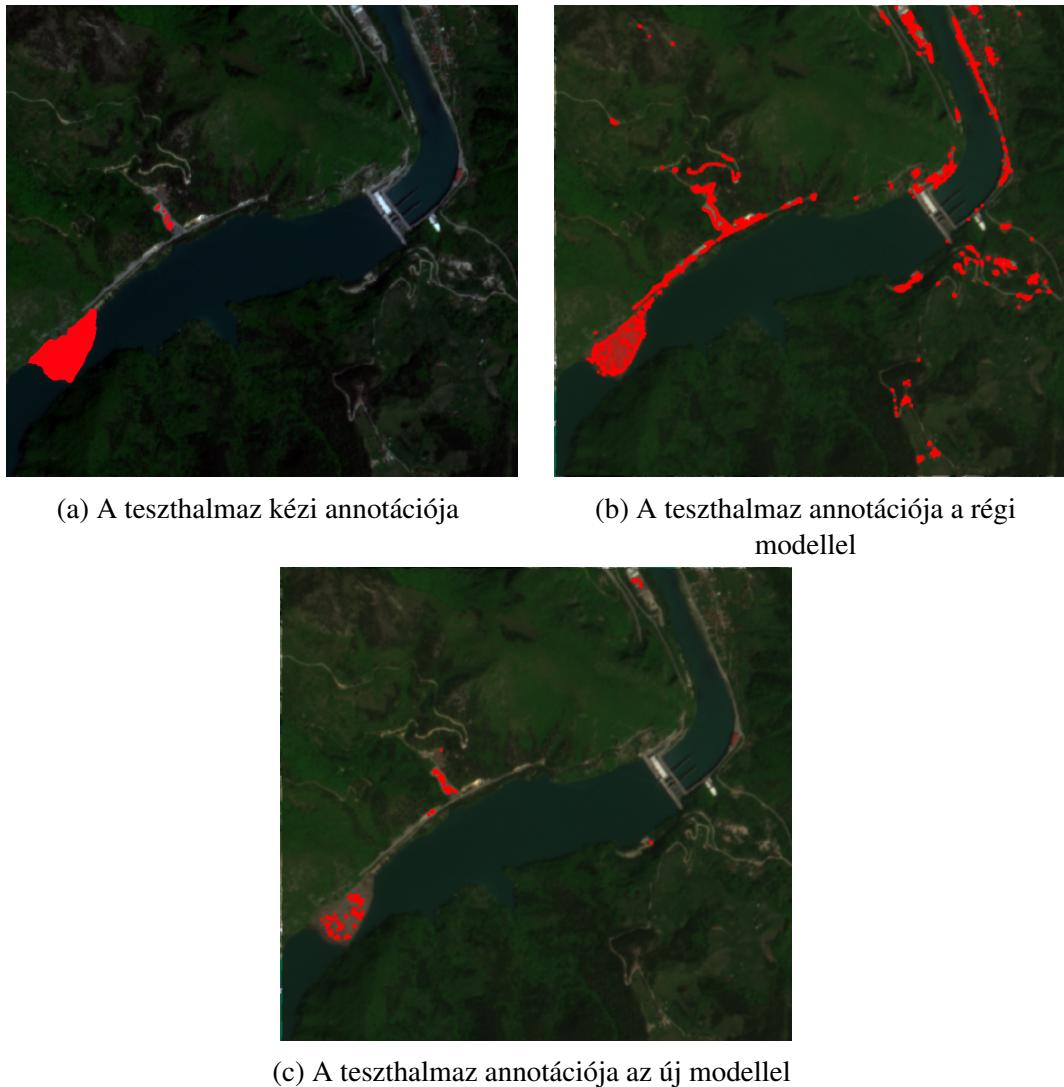
A teszthalmaz három drinai felvételből áll, amin kézzel annotáltam a hulladékkal szennyezett területeket. Ez a terület egyben egy szárazföldi hulladéklerakót 5.2, illetve egy vízfelszíni hulladékszigetet is tartalmaz, így alkalmas minden két detektálásnak a tesztelésére. A 5.1 ábrából látható, hogy Drinán úgy fogják meg az úszó műanyag-alapú hulladékot, hogy egy zsinorra röhíznak üres hordókat, melyek a víz felszínén lebegnek. Így, minden, ami elég könnyű ahhoz, hogy a folyó felszínen ússzon (műanyagpalackok, kisebb fadarabok) megakad a hordók mögött, míg például nagyobb fadarabok, vagy más, nehezebb uszadékok a zsinor alatt elúsznak. Így a folyó felszínén kialakuló sziget nagy koncentrációban tartalmaz műanyag alapú hulladékot, tehát alkalmas arra, hogy a modellt ezen validáljam vízfelszíni hulladékdetectáláshoz. Ráadásul erről a területről nem készültek tesztadatok ebben a kutatásban, így a modell teljesítménye az itteni felvételeken jól tesztelhető. A 5.3 ábrából látható egy-egy vizuális összehasonlítás a régi és az új modell klasszifikációja között a teszthalmaz egyik felvételén. Látszik ezen a példán, hogy az új modell több false negative-ot termel főleg a hulladéksziget körül, de ugyanakkor lényegesen lecsökkenti a false positive-ok arányát a régi modellhez képest. Ráadásul a folyó mellett található hulladéklerakót is megtalálja az új modell, míg a régi modell nem találja meg, ellenben a lerakó környékét és az utakat, épületeket gyakran hulladéknak detectálja. Ez egy fontos eredmény, hiszen amint a 3.1 fejezetben is tárgyaltam, célja ennek a kutatásnak, hogy csökkentsem a modell false positive arányait, miközben továbbra is meg tudja találni a hulladéklerakókat, illetve hulladékszigeteket.



5.1. ábra. A drinai hulladéksziget. Egy lebegő zsinor fogja meg a műanyagpalackokat [32]



5.2. ábra. A Drina melletti féllegális szemétlerakó [33]



5.3. ábra. Az új modell összehasonlítása a régi modellel az egyik teszt felvételen

5.2. Teljesítmény mérési módszerek

A teszthalmaz eredményekeit a "Confusion Matrix" módszerével értékeltem ki [34]. Ezután ezeket az értékeket arra használtam, hogy a "Comission rate" (5.1 képlet), "Omission rate" (5.2 képlet), "Match rate" (5.3 képlet), illetve "Extraction rate" (5.4 képlet) értékeket számítsam ki [35].

$$\text{Comission rate} = \frac{N_{com}}{N_{ref}} \quad (5.1)$$

$$\text{Omission rate} = \frac{N_{om}}{N_{ext}} \quad (5.2)$$

$$Match\ rate = \frac{N_{match}}{N_{ref}} \quad (5.3)$$

$$Extraction\ rate = \frac{N_{ext}}{N_{ref}} \quad (5.4)$$

N_{com} , N_{om} , N_{match} , N_{ext} , N_{ref} , rendre a false positive, false negative (A mátrix mellékátlói), true positive (A mátrix főátlója), a modell által detektált pozitív, illetve a referencia adatokban található pozitív értékek. Összehasonlítottam az új modell teljesítményét a régi modell teljesítményével. A 5.1 táblázatból látható a két modell teljesítményének az átlaga a három felvételen.

Mérés azonosító	Régi modell átlagai (%)	új modell átlagai (%)
Comission Rate	63.67	28.13
Omission Rate	26.21	70.67
Match Rate	73.79	29.32
Extraction Rate	208.18	41.31

5.1. táblázat. A régi modell és az új modell teszteredményei átlagolva

Az új modell egy jóval kisebb false positive aránnyal rendelkezik mint a régi modell, de cserében a false-negative arányok is nagyok. Ennek oka a 5.3 ábrából látható, hiszen a régi modell sokkal több pontot detektál a hulladékszigeten, míg az új modell kevesebb pontot detektál, de továbbra is nagy mértékben megtalálja a hulladékszigetet. Illetve a 5.1 fejezetben is tárgyaltam, hogy a szárazföldi hulladéklerakót a folyó mellett az új modell már megtalálja, míg a régi nem találja meg. Tekintve arra, hogy a match rate a true positive-al arányos, és az Extraction rate az összes pozitive-al arányos, ezek az értékek is kisebbek lesznek, mint a régi modell értékei.

6. fejezet

Megvalósítás és alkalmazás

6.1. A meglevő alkalmazások bővítése

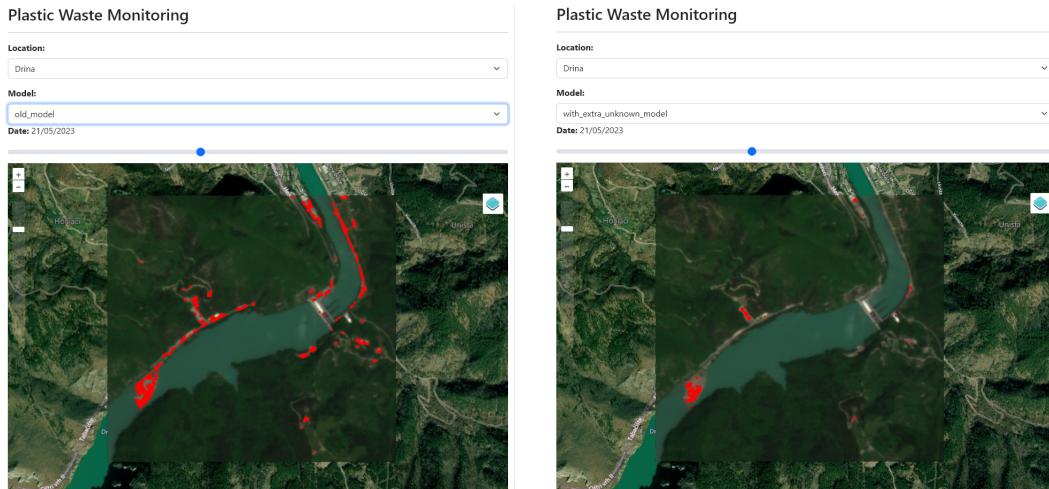
6.1.1. Az asztali alkalmazás bővítése

A meglevő asztali alkalmazás alkalmas volt a tanítóadatok hatékony előállítására, de utólag nem lehetett visszanézni, hogy adott műholdfelvételhez milyen tanítóadatok tartoznak, illetve azt sem, hogy az adott tanítóadat hol volt mintavételezve. Az alkalmazás eredetileg egy CSV fájlban [36] tárolta el az összes pixel spektrális értékeit és indexeit, és ezt lehetett használni tanításra. Ennek az volt a hátránya, hogy nehéz volt áttekinteni illetve kiegészíteni az adatokat. Ezért az asztali alkalmazást kiegészítettem ezzel a funkcionálitással, a tanítóadatok előállítása elmentésekor az alkalmazás létrehoz egy külön raszteres réteget is külön minden műholdfelvételhez, melyen látható, hogy mely területek voltak hozzáadva a tanítóadatok közé, így tetszőleges módon előállítható/ellenőrizhető a tanítóhalmaz.

6.1.2. A szerveralkalmazás bővítése

A szerveralkalmazás és webalkalmazás is bővítésre került: a szerveralkalmazás mostmár több modellt is le tud futtatni a letöltött műholdfelvételeken és ezeket külön tárolja. A webalkalmazás mostmár képes letölteni külön ezeket az eredményeket és több hulladékmászkoló módszer eredményét is meg tudja jeleníteni, ennek köszönhetően ezeket egymással össze lehet könnyen hasonlítani valós tesztadatokon. A 6.1 ábrán látható, ahogy lehet váltakozni a hulladékdetektálási modellek között.

Melyik linket
írjam ide, mit
írjak szerzőnek?



(a) A webalkalmazás a régi modell eredményeit mutatja a Drinán

(b) A webalkalmazás az új modell eredményeit mutatja a Drinán

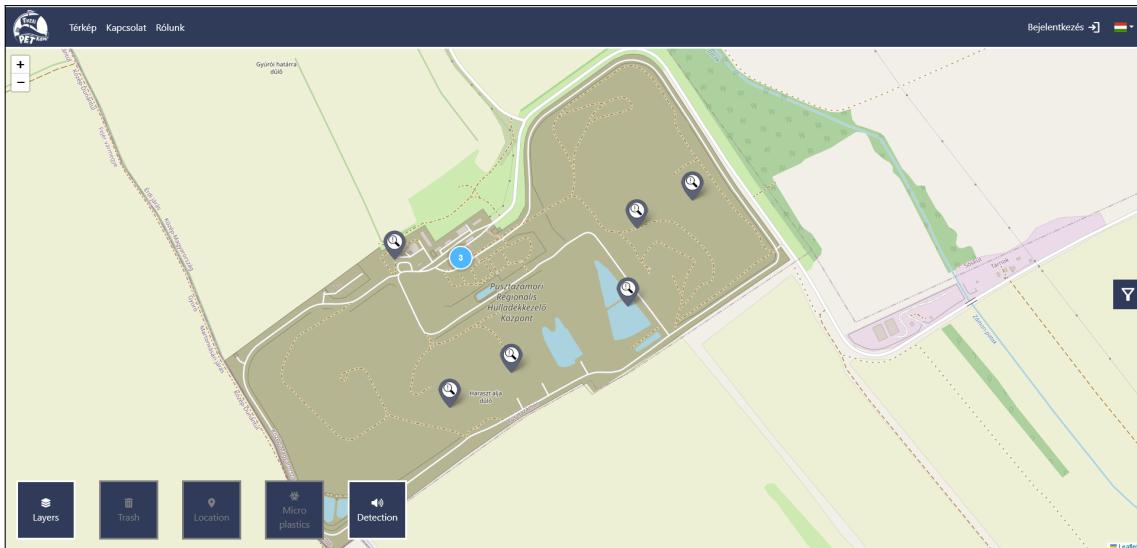
6.1. ábra. A webalkalmazás képes több modellnek az eredményét is összehasonlítóképpen megmutatni

6.2. A Tisza-Tisza alkalmazás

melyik link kerüljön ide?

A Tisza-Tisza webalkalmazás a PET Kupa által használt webalkalmazás, melynek az a célja, hogy egy olyan felületet biztosítson, ahol meglehet tekinteni a jelenleg ismert folyómentén található hulladéklerakókat, illetve akár a regisztrált felhasználók is be tudnak jelenteni ilyet. A PET Kupa megbízta az egyetemet azzal a feladattal, hogy ezt továbfejlessze, és a feladatok közé tartozott az is, hogy a Random Forest modell eredményeit integráljuk ebbe az alkalmazásba. Ezt a feladatot én vállaltam el.

Tekintve arra, hogy a Tisza-Tisza térképen pontok vannak megjelenítve, a modell által detektált területeket is pontokkal jelöljük. Ehhez egy nagyobb terület közepére helyezünk el egy pontot. Előfordulhat olyan is, hogy a modell olyan képeket klasszifikál, melyek el vannak torzítva (például magas párata tartalom miatt). Ilyenkor a false-positive-ok aránya lényegesen megnő. Ennek korrigálására a Tisza-Tisza alkalmazásban a legutolsó három detektálást (legfeljebb 1 hónap különbséggel) veszem figyelembe és két kép közös metszetével döntöm el, hogy milyen területek kerülnek fel a térképre. A 6.2 ábrán láthatóak a hulladékdetektálás során előállított pontok a térképen, melyeket egy nagyítóval vannak jelölve, melyben van egy felkiáltójel. A Hulladékdetektálás konverzióját a térképen látható pontokká a 6.3 fejezetben részletezem.



6.2. ábra. A hulladékdetectálás által megjelenített pontok a térképen.

6.2.1. A képfeldolgozás gyorsítása

A meglevő képfeldolgozó algoritmuson gyorsítani kellett, hogy elfogadható időn belül tudja feldolgozni a tanítóadatokat, illetve a naponta letöltött műholdfelvételeket a szerveren. Ezt a kutatólabor korábbi cikkjében párhuzamosítással javasolták, de egy egyszálú megoldással lényegesen tudtam gyorsítani a képfeldolgozásban: Ehhez egy nagyon hatékony Python programcsomagot, a Numpy-t [37] használtam fel, mellyel lényegesen megnövelte a feldolgozás sebességét: A tanítóadatok feldolgozásakor a régi módszer (6.1 forráskód) 16 felvételt tudott feldolgozni 4 nap és 10 óra alatt, míg az átírt módszer (6.2 forráskód) feldolgozott 85 felvételt 20 perc alatt.

```

1 def _calculate_index(numerator: np.ndarray, denominator: np.ndarray
2 ) -> np.ndarray:
3     """
4         Calculating an index based on given numerator and denominator.
5
6         :param numerator: numerator matrix
7         :param denominator: denominator matrix
8         :return: result matrix, containing the calculated values
9     """
10
11     # variables
12     rows = numerator.shape[0]
13     cols = numerator.shape[1]
14     index = np.ndarray(

```

```

14         shape=numerator.shape,
15         dtype="float32",
16     )
17
18     # calculate index
19     for i in range(rows):
20         for j in range(cols):
21             if np.isnan(numerator[i, j]) or np.isnan(denominator[i,
22                         j]):
23                 index[i, j] = float("NaN")
24             elif denominator[i, j] != 0:
25                 index[i, j] = numerator[i, j] / denominator[i, j]
26             else:
27                 if numerator[i, j] < 0:
28                     index[i, j] = np.nanmin(numerator)
29                 elif numerator[i, j] > 0:
30                     index[i, j] = np.nanmax(numerator)
31                 else:
32                     index[i, j] = float("NaN")
33
34     # return index values
35     return index

```

6.1. forráskód. A képfeldolgozás régi módszere

```

1 def calculate_index(numerator: np.ndarray, denominator: np.ndarray)
2     -> np.ndarray:
3
4     """
5         Calculating an index based on given numerator and denominator.
6
7         :param numerator: numerator matrix
8         :param denominator: denominator matrix
9         :return: result matrix, containing the calculated values
10    """
11
12    # variables
13    index = np.ndarray(
14        shape=numerator.shape,
15        dtype="float32",
16    )

```

```

16     numerator_nan_min = np.nanmin(numerator)
17     numerator_nan_max = np.nanmax(numerator)
18
19     # calculate index
20     nan_mask = np.isnan(numerator) | np.isnan(denominator)
21     numerator_zero_mask = numerator == 0
22     denominator_zero_mask = denominator == 0
23
24     invalid_mask = nan_mask | (numerator_zero_mask &
25                               denominator_zero_mask)
25     valid_mask = np.logical_not(invalid_mask)
26
27     valid_denominator_non_zero_mask = valid_mask & np.logical_not(
28                               denominator_zero_mask)
28     valid_denominator_zero_mask = valid_mask &
29                               denominator_zero_mask
29
30     numerator_positive_denominator_zero_mask =
31                               valid_denominator_zero_mask & (numerator > 0)
31     numerator_negative_denominator_zero_mask =
32                               valid_denominator_zero_mask & (numerator < 0)
32
33     index[invalid_mask] = float("NaN")
34     index[numerator_positive_denominator_zero_mask] =
35                               numerator_nan_max
35     index[numerator_negative_denominator_zero_mask] =
36                               numerator_nan_min
36     index[valid_denominator_non_zero_mask] = (
37         numerator[valid_denominator_non_zero_mask] / denominator[
38                               valid_denominator_non_zero_mask]
38     )
39
40     # return index values
41     return index

```

6.2. forráskód. A képfeldolgozás numpy-al

[5]-ban megmérték a modell feldolgozási sebességét különböző méretű felvételeken. Az első felvételt el tudtam kérni a szerzőtől, de a többi felvételt újból előállítottam, igyekezve arra, hogy hasonló legyen a felvételek mérete. A 6.1 táblázatban találhatóak a régi feldolgozás sebességei a régi modell mellett. A 6.2 táblázatban látható a régi

új módszert is megjeleníteni

és új modell feldolgozási sebessége az új számolási módszerrel. Fontos megjegyezni azt, hogy nem ugyanazon a számítógépen történt az új módszer sebességének a lemerése, de a két számítógép számolóképessége ugyanabban a súlycsoportban van. A két táblázatból láthatjuk, hogy az új indexszámolási módszer körülbelül felére csökkentette a feldolgozási és osztályozási időt, mostmár az idő nagy része az osztályozással és az osztályozás feldolgozásával megy el. További kutatás esetén érdemes lehet megvizsgálni az osztályozás feldolgozásával kapcsolatos lassulásokat annak érdekében, hogy ezt a feldolgozást gyorsítsuk.

Felvétel mérete (pixel)	feldolgozási idő
$164 \times 312 = 51\ 168$	2.17 mp
$1194 \times 801 = 956\ 394$	39.98 mp
$4597 \times 4153 = 19\ 091\ 341$	12 perc 42 mp
$6614 \times 5981 = 39\ 558\ 334$	26 perc 5 mp

6.1. táblázat. A régi modell régi módszerrel való feldolgozási ideje

Felvétel mérete (pixel)	Feldolgozási idő (Régi modell)
$164 \times 312 = 51\ 168$	1.88mp
$1262 \times 820 = 1\ 034\ 840$	19.06 mp
$4951 \times 4002 = 19\ 813\ 902$	6 perc 16 mp
$10366 \times 3860 = 40\ 012\ 760$	14 perc 24 mp

6.2. táblázat. A régi és új modell új módszerrel való feldolgozási ideje

6.3. Közös metszet

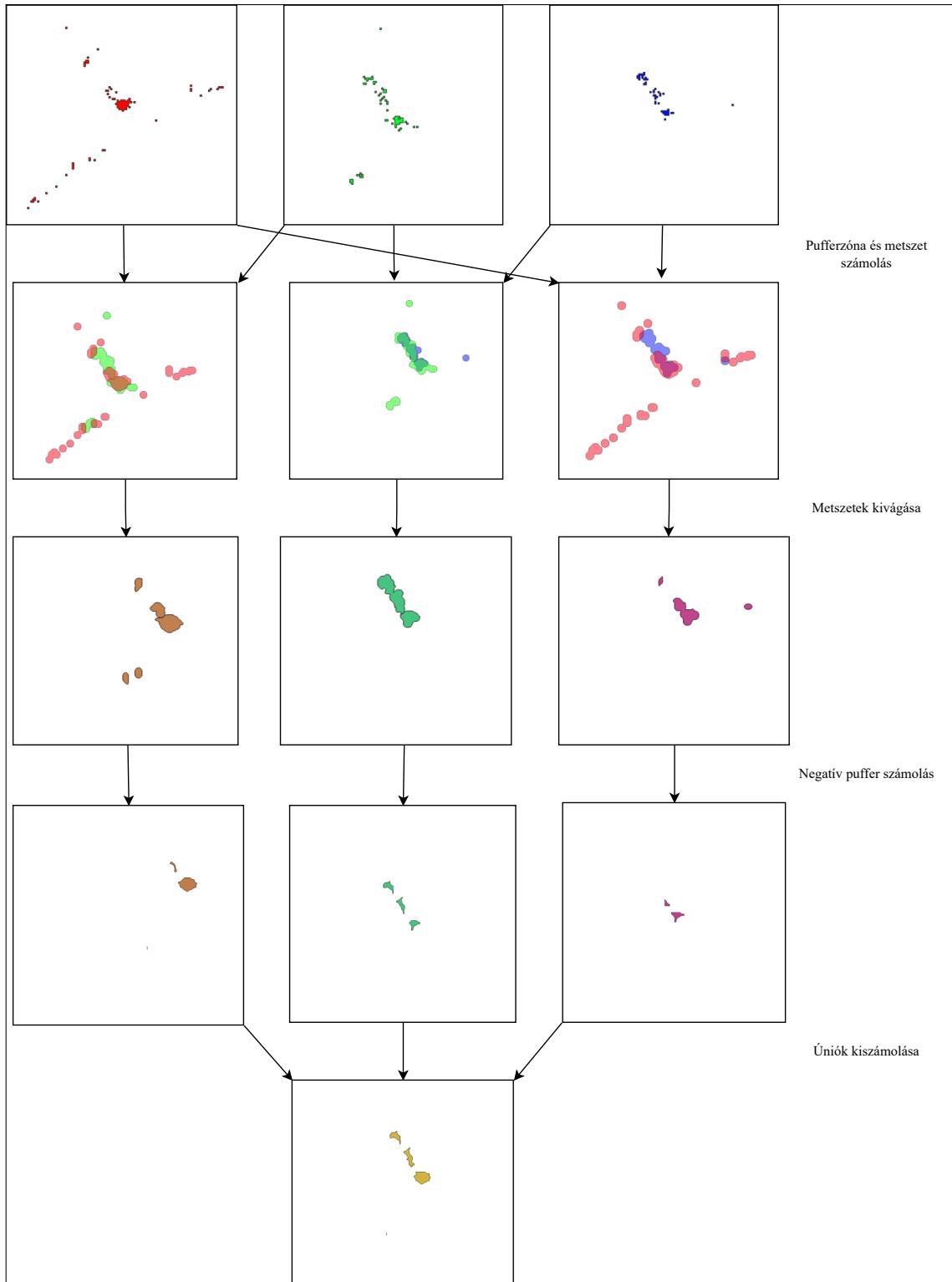
A már meglevő szervertől poligonok formájában, GeoJSON-ben [38] lehet lekérni az adott napon detektált hulladékos területeket. így érdemes poligonok metszetében kigondolni a többségi szavazást. Jelöljük BUF(P, n)-vel egy multipolygon pufferét, ahol $P \in \mathbb{P}$ egy multipolygon, és n egy egész szám. Ekkor a többségi szavazást három

képre a 6.1 képlet szerint lehet alkalmazni. Ezután a poligonok egy-egy belső pontját megválasztva megtudjuk jelölni a hulladéklerakókat. A megközelítés szavakra bontva a következő: az összes multipolygon párra vesszük a két multipolygon metszetét, (ahol egy n méretű hibahatár is belefér a metszet számolásba), utána a poligonpárokat rendre összeuniózzuk. A 6.3 ábra vizualizálja a képlet lépésein. Ezzel a módszerrel ki lehet szűrni azokat a hibákat, melyek rossz minőségű felvételek miatt keletkeznek. Tekintve arra, hogy nem minden Planet felvétel kerül osztályozásra, mivel előfordulnak hiányos vagy

hasonló
képleteket
mások is
ismernek
fórumokban,
de hivatalos
forrással nem
találkoztam

felhős felvételek, érdemes megszabni egy időintervallumot, amin belül alkalmazható a többségi szavazás. A dolgozat írásának idejében legfeljebb 1 hónap intervallumon el kell helyezkedjen minden a három felvétel.

$$Eredmény\ multipolygon = \bigcup_{P_1 \in \mathbb{P}} \bigcup_{P_2 \in \mathbb{P}} BUF(BUF(P_1, n) \cap BUF(P_2, n), -n) \quad (6.1)$$



6.3. ábra. A 6.1 képlet lépései

7. fejezet

Összefoglalás és eredmények

Köszönetnyilvánítás

Amennyiben a TDK projekted pénzügyi támogatást kapott egy projektből vagy az egyetemtől, jellemzően kötelező feltüntetni a dolgozatban is. A dolgozat elkészítéséhez segítséget nyújtó oktatók, hallgatótársak, kollégák felé is nyilvánítható külön köszönet.

A. függelék

Szimulációs eredmények

Irodalomjegyzék

- [1] M.G. Kibria, N.I. Masuk és R. et al. Safayet. „Plastic Waste: Challenges and Opportunities to Mitigate Pollution and Effective Management”. *International Journal of Environmental Research* 17.20 (2023. jan.). ISSN: 2008-2034. URL: <https://doi.org/10.1007/s41742-023-00507-z>.
- [2] Björn Nyberg és tsai. „Leaving a plastic legacy: Current and future scenarios for mismanaged plastic waste in rivers”. *Science of The Total Environment* 869 (2023), 161821. old. ISSN: 0048-9697. DOI: <https://doi.org/10.1016/j.scitotenv.2023.161821>. URL: <https://www.sciencedirect.com/science/article/pii/S0048969723004369>.
- [3] Tim H. M. van Emmerik és tsai. „River plastic transport and deposition amplified by extreme flood”. *Nature Water* 1.6 (2023), 514–522. old. ISSN: 2731-6084. DOI: [10.1038/s44221-023-00092-7](https://doi.org/10.1038/s44221-023-00092-7). URL: <https://doi.org/10.1038/s44221-023-00092-7>.
- [4] Leo Breiman. „Random Forests”. *Machine Learning* 45.1 (2001), 5–32. old. ISSN: 1573-0565. DOI: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324). URL: <https://doi.org/10.1023/A:1010933404324>.
- [5] Dávid Magyar és tsai. „Waste Detection and Change Analysis based on Multispectral Satellite Imagery”. 2023. jan. DOI: [10.48550/arXiv.2303.14521](https://arxiv.org/abs/2303.14521).
- [6] Anjar Dimara Sakti és tsai. „Identification of illegally dumped plastic waste in a highly polluted river in Indonesia using Sentinel-2 satellite imagery”. *Scientific Reports* 13.1 (2023), 5039. old. ISSN: 2045-2322. DOI: [10.1038/s41598-023-32087-5](https://doi.org/10.1038/s41598-023-32087-5). URL: <https://doi.org/10.1038/s41598-023-32087-5>.
- [7] Gil Gonçalves és Umberto Andriolo. „Operational use of multispectral images for macro-litter mapping and categorization by Unmanned Aerial Vehicle”. *Marine Pollution Bulletin* 176 (2022), 113431. old. ISSN: 0025-326X. DOI: <https://doi.org/10.1016/j.marpolbul.2022.113431>.

- doi.org/10.1016/j.marpolbul.2022.113431. URL: <https://www.sciencedirect.com/science/article/pii/S0025326X22001138>.
- [8] Antonio Lanorte és tsai. „Agricultural plastic waste spatial estimation by Landsat 8 satellite images”. *Computers and Electronics in Agriculture* 141 (2017), 35–45. old. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2017.07.003>. URL: <https://www.sciencedirect.com/science/article/pii/S0168169917302259>.
 - [9] Dan Zeng és tsai. „Multi-Scale CNN Based Garbage Detection of Airborne Hyperspectral Data”. *IEEE Access* 7 (2019), 104514–104527. old. DOI: 10.1109/ACCESS.2019.2932117.
 - [10] Xian Sun és tsai. „Revealing influencing factors on global waste distribution via deep-learning based dumpsite detection from satellite imagery”. *Nature Communications* 14.1 (2023), 1444. old. ISSN: 2041-1723. DOI: 10.1038/s41467-023-37136-1. URL: <https://doi.org/10.1038/s41467-023-37136-1>.
 - [11] Rocio Nahime Torres és Piero Fraternali. „Learning to Identify Illegal Landfills through Scene Classification in Aerial Images”. *Remote Sensing* 13.22 (2021). ISSN: 2072-4292. DOI: 10.3390/rs13224520. URL: <https://www.mdpi.com/2072-4292/13/22/4520>.
 - [12] Kaiming He és tsai. „Deep Residual Learning for Image Recognition”. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
 - [13] Tsung-Yi Lin és tsai. „Feature Pyramid Networks for Object Detection”. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.
 - [14] Robert Page és tsai. „Identification of Tyre and Plastic Waste from Combined Copernicus Sentinel-1 and -2 Data”. *Remote Sensing* 12.17 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12172824. URL: <https://www.mdpi.com/2072-4292/12/17/2824>.
 - [15] Nicolò Taggio és tsai. „A Combination of Machine Learning Algorithms for Marine Plastic Litter Detection Exploiting Hyperspectral PRISMA Data”. *Remote*

- Sensing* 14.15 (2022). ISSN: 2072-4292. DOI: 10 . 3390 / rs14153606. URL: <https://www.mdpi.com/2072-4292/14/15/3606>.
- [16] Mattis Wolf és tsai. „Machine learning for aquatic plastic litter detection, classification and quantification (APLASTIC-Q)”. *Environmental Research Letters* 15.11 (2020), 114042. old. DOI: 10 . 1088 / 1748 - 9326 / abbd01. URL: <https://dx.doi.org/10.1088/1748-9326/abbd01>.
 - [17] Gil Gonçalves és tsai. „Mapping marine litter using UAS on a beach-dune system: a multidisciplinary approach”. *Science of The Total Environment* 706 (2020), 135742. old. ISSN: 0048-9697. DOI: <https://doi.org/10.1016/j.scitotenv.2019.135742>. URL: <https://www.sciencedirect.com/science/article/pii/S0048969719357377>.
 - [18] Ousmane Youme és tsai. „Deep Learning and Remote Sensing: Detection of Dumping Waste Using UAV”. *Procedia Computer Science* 185 (2021). Big Data, IoT, and AI for a Smarter Future, 361–369. old. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2021.05.037>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050921011224>.
 - [19] Barry de Ville. „Decision trees”. *WIREs Computational Statistics* 5.6 (2013), 448–455. old. DOI: <https://doi.org/10.1002/wics.1278>. eprint: <https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.1278>. URL: <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1278>.
 - [20] Jr Harrel. URL: <https://lib.stat.cmu.edu/S/Harrell/data/descriptions/titanic.html> (elérés dátuma 2013. 09. 23.).
 - [21] IBM. URL: <https://www.ibm.com/topics/random-forest> (elérés dátuma 2024. 05. 05.).
 - [22] Planet Labs. URL: <https://www.planet.com/company/> (elérés dátuma 2024. 05. 06.).
 - [23] European Space Agency. URL: <https://earth.esa.int/eogateway/missions/planetscope> (elérés dátuma 2024. 05. 06.).
 - [24] Planet Labs. URL: <https://developers.planet.com/docs/apis/data/sensors/> (elérés dátuma 2024. 04. 03.).

- [25] Kyriacos Themistocleous és tsai. „Investigating Detection of Floating Plastic Litter from Space Using Sentinel-2 Imagery”. *Remote Sensing* 12.16 (2020). ISSN: 2072-4292. DOI: 10.3390/rs12162648. URL: <https://www.mdpi.com/2072-4292/12/16/2648>.
- [26] S. K. McFEETERS. „The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features”. *International Journal of Remote Sensing* 17.7 (1996), 1425–1432. old. DOI: 10.1080/01431169608948714.
- [27] InfoCons.ro. URL: <https://fiiunexemplu.ro/in-romania-exista-46-depozite-de-deseuri-gropi-de-gunoii/> (elérés dátuma 2024. 04. 02.).
- [28] Google. URL: <https://www.google.com/maps/> (elérés dátuma 2024. 05. 01.).
- [29] Hervé Abdi és Lynne J. Williams. „Principal component analysis”. *WIREs Computational Statistics* 2.4 (2010), 433–459. old. DOI: <https://doi.org/10.1002/wics.101>. eprint: <https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.101>. URL: <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.101>.
- [30] Tom Howley és tsai. „The Effect of Principal Component Analysis on Machine Learning Accuracy with High Dimensional Spectral Data”. 2005. jan., 209–222. old. ISBN: 978-1-84628-223-2. DOI: 10.1007/1-84628-224-1_16.
- [31] F. Pedregosa és tsai. „Scikit-learn: Machine Learning in Python”. *Journal of Machine Learning Research* 12 (2011), 2825–2830. old.
- [32] EuroNews. URL: <https://www.euronews.com/green/2023/06/04/we-will-die-before-the-drina-is-clean-bosnia-villagers-hope-for-a-solution-to-polluted-riv> (elérés dátuma 2024. 05. 05.).
- [33] Pet Kupa. URL: <https://www.petkupa.hu/> (elérés dátuma 2024. 05. 08.).
- [34] Russell G. Congalton. „A review of assessing the accuracy of classifications of remotely sensed data”. *Remote Sensing of Environment* 37.1 (1991), 35–46. old. ISSN: 0034-4257. DOI: [https://doi.org/10.1016/0034-4257\(91\)90048-B](https://doi.org/10.1016/0034-4257(91)90048-B). URL: <https://www.sciencedirect.com/science/article/pii/003442579190048B>.

- [35] Anett Fekete és Mate Cserep. „Tree segmentation and change detection of large urban areas based on airborne LiDAR”. *Computers & Geosciences* 156 (2021), 104900. old. ISSN: 0098-3004. DOI: <https://doi.org/10.1016/j.cageo.2021.104900>. URL: <https://www.sciencedirect.com/science/article/pii/S0098300421001928>.
- [36] Yakov Shafranovich. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. RFC 4180. 2005. okt. DOI: 10.17487/RFC4180. URL: <https://www.rfc-editor.org/info/rfc4180>.
- [37] Charles R. Harris és tsai. „Array programming with NumPy”. *Nature* 585.7825 (2020. szept.), 357–362. old. DOI: 10.1038/s41586-020-2649-2. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [38] H. Butler és tsai. *The GeoJSON Format*. RFC 7946. 2016. aug. DOI: 10.17487/RFC7946. URL: <https://www.rfc-editor.org/info/rfc7946>.

Ábrák jegyzéke

2.1.	Hulladékdetektálás "Adjusted Plastic Index", Random Forest és Mahalanobis távolság segítségével [6]	6
2.2.	Spectral angle mapping referencia adatainak előállítása [7]	7
2.3.	Az MSCNN működési elve [9]	8
2.4.	Az BCA-Net működési elve [10]	9
2.5.	A tengeri hulladékdetektáláshoz való tanítóadatok előállítása [15]	10
2.6.	A gépi tanulás, geomorfológia, fotogammetria és hidrodinamikai modellezés használása hulladékdetektálásra [17]	11
2.7.	A döntési fa működési elve elve [19]	12
2.8.	A Random Forest működési elve [21]	13
3.1.	Különböző indexek összehasonlítása egy mesterségesen előállított úszó műanyag-palack szigeten [25]	17
4.1.	A műanyag alapú hulladék törmelék mellé helyezve. Forrás: Google Maps	19
4.2.	A csökkentett modell hasonlóan teljesít a teljes méretű modellhez	20
4.3.	Az adatok közötti aránytalanság, logaritmikus skálázással	21
4.4.	Nyári és téli adatok összehasonlítása	23
5.1.	A drinai hulladéksziget. Egy lebegő zsinor fogja meg a műanyagpalackokat [32]	25
5.2.	A Drina melletti féllegális szemétlerakó [33]	25
5.3.	Az új modell összehasonlítása a régi modellel az egyik teszt felvételen . .	26
6.1.	A webalkalmazás képes több modellnek az eredményét is összehasonlítképpen megmutatni	29
6.2.	A hulladékdetektálás által megjelenített pontok a térképen.	30
6.3.	A 6.1 képlet lépései	35

Táblázatok jegyzéke

2.1. A PSB.SD műszer hullámhosszai [24]	14
4.1. A tanítóadatok címkéi	19
4.2. A döntési fák méretének a mediánja nem csökkent, amint a főkomponensek száma csökkent, cserében 3 főkomponensnél már nőtt. . .	21
5.1. A régi modell és az új modell teszteredményei átlagolva	27
6.1. A régi modell régi módszerrel való feldolgozási ideje	33
6.2. A régi és új modell új módszerrel való feldolgozási ideje	33

Algoritmusjegyzék

Forráskódjegyzék

6.1. A képfeldolgozás régi módszere	30
6.2. A képfeldolgozás numpy-al	31